UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P O Box 1450
Alexandria, Virginia 22313-1450
www uspto gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/587,907	07/27/2006	Claude Lamblin	P1923US	3511

8968 7590 05/12/2008
DRINKER BIDDLE & REATH LLP
ATTN: PATENT DOCKET DEPT.
191 N. WACKER DRIVE, SUITE 3700
CHICAGO, IL 60606

EXAMINER	
BORSETTI, GREG	

ART UNIT	PAPER NUMBER
4141	

MAIL DATE	DELIVERY MODE
05/12/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/587,907	LAMBLIN ET AL.	
	Examiner	Art Unit	
	GREG A. BORSETTI	4141	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 July 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5,7-14,18-20 and 22-31* is/are rejected.

7)☒ Claim(s) *6,15-17 and 21* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *27 July 2006* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *12/22/2006*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to preliminary amendment filed on 7/27/2006.

2. Claims 1-4, 7-8, 10, 12-15, 18, 20-23, 25-31 have been amended.

3. Claims 1-31 are pending.

Information Disclosure Statement

4. The Information Disclosure Statement (IDS) submitted on 12/22/2006 is in compliance with the provisions of 37 CFR 1.97.

Drawings

5. The drawings filed on 7/27/2006 are accepted by the examiner.

Specification

6. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Claim Objections

7. Claims 1 and 22 are objected to because of the following informalities: The claim use ambiguous terminology in “on one hand…on the other hand”. Furthermore, the structure of the claim is incorrect. The dictionary of codevectors comprises inter-

embedded dictionaries and for a given dimensions a union wherein... of a first and a second set. Appropriate correction is required.

Claim 23 is objected to because of the following informalities: The claim further limits, in the preamble, to include a memory as a part of the method. Claim 4 claims no such memory. Appropriate correction is needed. Furthermore, the claim uses ambiguous terminology in “once and for all…” Appropriate correction is needed.

Claims 4,8,11,13-14,20,22-31 are objected to for using shorthand in the instances of “lower/higher” and “inserting/deleting…etc.” Correction is suggested. For the purposes of examination the example phrase "lower/higher" s considered to be "lower or higher."

Claim 6 says “up to said given dimension N" where N is lower than the initial dictionary dimension n. It should say "down to said dimension 1." Correction is needed.

Claims 5-6 recite "n+i" and "n-i." “i” is not defined in the claim. Correction is needed.

Claim 7 refers to 2 different "n" values which should be made clearer. Correction is needed.

Claim 11 is objected to for having a misspelling. “modelled" should be rewritten "modeled." Correction is needed.

Claims 6, 15, 16, 17, and 21 are objected to for having allowable subject matter but are dependent upon rejected claim 4.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claims 5-7 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The cited claims use a variable ***"i"*** which is not defined in the claim. Correction is needed.

Claim 7 is rejected under 35 U.S.C. 112, second paragraph as lacking antecedent basis. The claim points to two different "n" values determined by the steps a1)-a3) and a'1)-a'3). However, claim 7 depends on claim 5 and uses steps a'1)-a'3) from claim 6 to derive its result.

Claim 11 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The claim cites that the source is modeled by a "learning sequence." It is not completely understood what is meant by a learning sequence and it is not described within the specification. For the purposes of examination, it will be interpreted as a statistical analysis.

Claim Rejections - 35 USC § 101

9. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 1-3 of the claimed invention are directed to non-statutory subject matter. Claims 1-3 claim a dictionary that does not provide an output or functional change to cause a useful, tangible, or concrete result which satisfies 35 U.S.C. 101. Correction is needed.

Claims 25-29 is directed to non-statutory subject matter. The claim does not provide an output or functional change to cause a useful, tangible, or concrete result which satisfies 35 U.S.C. 101. Furthermore, the steps involved in the method are describe a program per se. Correction is needed.

Claims 30 and 31 are directed to non-statutory subject matter for both claiming a "computer program product." Although the computer program product is further mentioned to be stored, it is non-statutory because a computer program product is an algorithm. Examiner suggests claiming the "computer readable memory storing a computer program product."

Claim 31 of the claimed invention are directed to non-statutory subject matter. Claim 31 claims a computer program product stored on a memory that implements the use of a dictionary in compression which does not provide an output or functional change to cause a useful, tangible, or concrete result which satisfies 35 U.S.C. 101. Correction is needed.

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-3, 8-11, 25-27, 29, and 31 are considered to be unpatentable over Gersho et al. (US Patent #5890110 hereinafter Gersho)

As per claim 1, Gersho discloses:

- **for a given dimension, inter-embedded dictionaries of increasing resolution**
- **for a given dimension, a union:**
    - **of a first set consisting of codevectors constructed by inserting, into codevectors of dictionaries of lower dimension, elements taken from a finite set of real numbers according to a finite collection of predetermined insertion rules**
    - [Gersho, Fig. 7] discloses an example of how Gersho develops an extended vector by inserting zeros from the Q vector to change the dimensionality of the codeword vector. This is used in unison with the original vector to create the extended vector. The first set would be the Q vectors because it uses a finite set of real numbers and inserts them into the extended vector. This is repeated to redefine a codebook for a certain dimensionality.
    - **a second set consisting of codevectors that may not be obtained by insertion into codevectors of lower dimension of the elements of said finite set according to said collection of insertion rules.**
    - [Gersho, Fig. 7], as described above shows the development of an extended vector of increased dimensionality by combining the Q vector and the S vector to produce the overall extended vector. In this case the S vector (estimated SSV) is not obtained by the predetermined rules. It is

generated by a DFT (discrete Fourier transform) and further comprises the extended vector.

As per claim 2, claim 1 is incorporated and Gersho discloses:

- **said collection of insertion rules is formulated on the basis of elementary rules consisting in inserting a single element of the finite set of real numbers in the guise of component at a given position of a vector**
- [Gersho, Fig. 7] discloses a vector Q which shows a positional relationship to the extended vector and determines the location of additions to the extended vector.

As per claim 3, claim 2 is incorporated and Gersho discloses:

- **each elementary rule is defined by a pair of two positive integers representative: of a rank of the element in said finite set, and of a position of insertion**
- [Gersho, Fig. 7] discloses the Q vector which is defined by 1's and 0's. The vector also further inherently determines position of insertion by weighting in the Q vector.

As per claim 8, claim 4 is incorporated and Gersho discloses:

- **said collection of insertion/deletion rules is formulated on the basis of elementary rules consisting in inserting/deleting a single element of the finite set of reals in the guise of component at a given position of a vector**
- [Gersho, Fig. 7] discloses a vector Q which shows a positional relationship to the extended vector and determines the location of additions to the extended vector.

As per claim 9, claim 8 is incorporated and Gersho discloses:

- **each elementary rule is defined by a pair of two positive integers representative: of a rank of the element in said finite set, and of a position of insertion/deletion**
- [Gersho, Fig. 7] discloses the Q vector which is defined by 1's and 0's. The vector also further inherently determines position of insertion by weighting in the Q vector.

As per claim 10, claim 4 is incorporated and Gersho discloses:

- **said finite set and said collection of insertion/deletion rules are defined a priori, before constructing the dictionary by analysis of a source to be quantized**
- [Gersho, column 7, lines 49-55] discloses how the vector Q is defined, which teaches the insertion rules because it determines where the values are inserted

into the extended vector X. Q is defined by the input pair of an estimated pitch value and a subvector and is done prior to constructing the dictionary.

As per claim 11, claim 10 is incorporated and Gersho discloses:

- **said source is modeled by a learning sequence and the definition of said finite set and of said collection of insertion/deletion rules is effected by statistical analysis of said source**
- [Gersho, column 7, lines 19-21] discloses "As mentioned earlier, S is assumed to have been sampled from some larger dimension random variable X, using the selector vector, Q." The spectral shape of S is modeled as shown at the top of Fig. 5, which is a learning sequence. The definition of the finite set and the collection of insertion or deletion rules is effected by the statistical analysis because the estimated spectral shape model is used in conjunction with the finite set and the collection of rules to define the extended vector as shown in Fig. 7.

As per claim 25, Gersho discloses:

- **The use of a nearest neighbor algorithm to compare the input vector to a codevector**
- [Gersho, column 7, lines 8-11] discloses "The best codevector in this new codebook which matches the input vector, ***S is selected as the representative by the nearest neighbor block***, 803."

- **CO2) at least on coding, calculation of a distance between the input vector and the codevector reconstituted in step CO1)**
- Gersho teaches a nearest neighbor calculation, where it would be inherent to calculate the distance between the input vector and the codevector.

Gersho does not specifically disclose,

- **CO1) for a current index (m.sup.j) of said codevector (x.sup.j) sought, reconstitution at least partial of a codevector of index (m') corresponding to said current index (m.sup.j), at least through the prior reading of the indices (j', m', l.sub.r) appearing in the correspondence tables making it possible to formulate said dictionary**
- As can be seen in the rejection of claim 24, it would be an obvious step in using pointers to reference indices instead of the information itself to reduce storage space. This would make it obvious in light of Gersho to reference the wanted information through its indices.
- **CO3) at least on coding, repetition of steps CO1) and CO2), for all the current indices in said dictionary**
- Gersho discloses a nearest neighbor calculation for which it would be inherent that all the codevectors in the dictionary would need to be compared to find the codevector with minimal distance. It would be obvious to apply this to the indices using pointers as described above.
- **CO4) at least on coding, identification of the index (m.sub.min) of the codevector at least partially reconstituted whose distance (d.sub.min),**

calculated in the course of one of the iterations of step CO2), with the input vector is the smallest

- Gersho discloses a nearest neighbor calculation for which it would be inherent that all the codevectors in the dictionary would need to be compared to find the codevector with minimal distance. It would be obvious to apply this to the indices using pointers as described above.
- **CO5) at least on decoding, determination of the nearest neighbour of the input vector (y) in the guise of codevector (x.sup.j) whose index (m.sub.min) has been identified in step CO4)**
- Gersho discloses a nearest neighbor calculation for which it would be inherent that all the codevectors in the dictionary would need to be compared to find the codevector with minimal distance. It would be obvious to apply this to the indices using pointers as described above.

Claim 26 is rejected under the same principles as claim 24 for having parallel limitations. Claim 26 claims the same subject matter but simply adds the reconstruction of the codevector to the wanted dimension, where it has been shown that the purpose of the insertion/deletion rules was for dimensionality changes.

As per claim 27, claim 25 is incorporated and Gersho teaches:

Gersho does not fully disclose,

- **CO11) the reading, in the correspondence tables, of indices representative of links to said second set and to the insertion/deletion rules and including: the index of a current dimension of a subset of said second set, the current index of an element of said subset, and the index of the appropriate insertion/deletion rule for the construction of the codevector of the dictionary of given dimension**
- It would be obvious to someone of ordinary skill in the art to characterize a codevector by index and index of its elements in a matrix. Furthermore, it would be obvious to someone of ordinary skill in the art to have a ruleset stored in one place that can be reached by a representative index of the wanted rule. Then, as described above, pointers could point from one index to another to gain the wanted result without actually storing the intermediary steps in order to effectively apply a rule to a codevector. Gersho discloses the use of a second set and rules to apply to that set where it would be obvious to someone of ordinary skill to use pointers, as in C++, to point to locations representing values and operations instead of actively defining them for every location for the purposes of lowering the storage requirements.
- **CO12) the reading, in the subset identified by its current dimension, of said element identified by its current index**
- It would be obvious that given a current dimension and the application of insertion/deletion rules that the index of the element in the given codevector would need to be read.

- **in step CO2), said distance is calculated as a function of a distortion criterion estimated as a function of: the index of the insertion/deletion rule, and of the element of the subset identified by its current index, thereby making it possible to only partially construct the codevector with said given dimension in step CO1), by reserving the complete reconstruction simply for decoding**
- [Gersho, claim 7] discloses "extracting from each of said codevectors a subcodevector of dimension L by selecting components of said codevector in accordance with said ordered set of index values; computing for each said subcodevector in said codebook a measure of distortion between said input subvector and said subcodevector." It has been shown the subcodevectors are developed as in Fig. 7 using the rules to modify the elements in the codevector. The extraction of elements from a codevector as shown in claim 7 teaches that the codevector is only partially constructed using the rules and elements. The decoding algorithm still uses the optimal codevector from the universal codebook. It would be obvious to someone of ordinary skill to use the indices instead of the information itself such as pointers (C++) to reduce storage requirements.

As per claim 29, claim 25 is incorporated and Gersho does not specifically teach:

- **at least said correspondence tables are stored in a memory of a coding/decoding device**

- [Gersho, column 11, lines 58-60] discloses “It can be easily integrated with other structured VQ approaches to customize the encoding/decoding to the need of the application in terms of complexity, memory, performance targets.” It would be obvious that if Gersho mentioned a memory that it can be assumed that Gersho's method can be embodied in a hardware embodiment or software embodiment with computer readable memory. It has been shown above that it would be obvious to someone of ordinary skill that the tables storing the information would be accessible through pointers such that the tables would only have to be stored in memory once.

Claim 31 is rejected under the same principles as claim 25 for having parallel limitations. Although claim 31 further defines a computer program product, [Gersho, column 11, lines 58-60] discloses “It can be easily integrated with other structured VQ approaches to customize the encoding/decoding to the need of the application in terms of complexity, memory, performance targets.” It would be obvious that if Gersho mentioned a memory that it can be assumed that Gersho's method can be embodied in a hardware embodiment with computer readable memory. It has been shown above that it would be obvious to someone of ordinary skill that the tables storing the information would be accessible through pointers such that the tables would only have to be stored in memory once.

Claims 4-5, 18-20, 22-24, and 30 are considered to be unpatentable over Gersho et al. (US Patent #5890110 hereinafter Gersho) in view of Chan et al. (NPL document "Constrained-Storage Vector Quantization in High Fidelity Audio Transform Coding" hereinafter Chan)

As per claim 4, Gersho discloses:

- **a first set consisting of codevectors formed by inserting/deleting into/from codevectors of dictionaries of lower/higher dimension elements taken from a finite set of real numbers according to a finite collection of predetermined insertion/deletion rules**
- [Gersho, Fig. 7] discloses an example of how Gersho develops an extended vector by inserting zeros from the Q vector to change the dimensionality of the codeword vector. This is used in unison with the original vector to create the extended vector. The first set would be the Q vector because it uses a finite set of real numbers and inserts them into the extended vector. This is repeated to redefine a codebook for a certain dimensionality using the predetermined insertion rules using the Q vector.
- **a first, intermediate, dictionary comprising at least said first set is constructed, for said given dimension**
- The codebook as disclosed by the iteration in operation from Fig. 7 is intermediate because it needs to calculate the entirety of the codebook in the changing of its dimensionality.

- **to adapt said dictionary to a use with at least one given resolution, a second, definitive, dictionary is constructed, on the basis of the intermediate dictionary,**
- [Gersho, Fig. 7] discloses an example of how Gersho develops an extended vector by inserting zeros from the Q vector to change the dimensionality of the codeword vector. This is used in unison with the original vector to create the extended vector. The first set would be the Q vector because it uses a finite set of real numbers and inserts them into the extended vector. Furthermore, [Gersho, column 8, lines 29-36] teaches "Equivalently, the encoder operation can be performed by constructing a new "codebook" by sub-sampling the universal codebook using Q to form a new set of codevectors called subcodevectors, having the same dimensionality L.sub.Q as the input variable dimension vector. Then, the encoder selects the subcodevector from this new codebook that best matches the input subvector." By sub-sampling, there would inherently be a deletion or removal of vector elements to reduce the dimensionality of the vector and this deletion would be done according to some predefined rules in order to retain as much information as possible when resampling to a lower dimension. The dictionary would inherently be adapted for use with at least one given resolution.

Gersho fails to teach:

- **by embedding/simplification of dictionaries of increasing/decreasing resolutions, the dictionaries of increasing resolutions being inter-**

embedded from the dictionary of smallest resolution up to the dictionary of greatest resolution

Chan, in analogous art, teaches the above limitation,

- [Chan, page 3598, column 1] discloses "if the features have the same dimension and the same or similar statistics, then they can jointly share a codebook. ***For variable rate coding, the shared codebook must also be an embedded or multi-resolution codebook*** with which different degrees of resolution can be attained to satisfy some bit allocation among the features." It would be obvious to embed/simplify dictionaries of different resolutions using the methods as disclosed in Gersho because Chan teaches that in variable rate encoding it would be inherent that the codebook would have to be embedded or multi-resolution.
- Chan and Gersho are analogous art because both pertain to variable dimension vector quantization. It would be obvious to someone of ordinary skill in the art to combine Chan with the Gersho device because [Chan, page 3597, column 1] discloses "We demonstrate that with CSVQ, tree structured codebooks can be constructed for very high rates without incurring an exponential growth in storage complexity and without impairing the rate-distortion performance. NLIVQ allows efficient coding of the power envelope, needed for transform coefficient normalization and adaptive distortion assignment. These techniques lead to a substantial reduction in the overall bit rate and codebook storage for

the audio coder." Chan provides a method for reducing storage complexity in vector quantization analogous to the instant application.

As per claim 5, claim 4 is incorporated and Gersho teaches:

- **an initial dictionary of initial dimension n, lower than said given dimension N, is obtained**
- [Gersho, Fig. 7] discloses a vector S which is lower in dimension than the extended vector X. This is an estimation for a single vector within a codebook but the process would be repeated for the plurality of the vectors within the codebook.
- **a first set consisting of codevectors of dimension n+i formed by inserting into codevectors of the initial dictionary elements taken from a finite set of real numbers according to a finite collection of predetermined insertion rules is constructed**
- [Gersho, Fig. 7] discloses an example of how Gersho develops an extended vector by inserting zeros from the Q vector to change the dimensionality of the codeword vector. This is used in unison with the original vector to create the extended vector. The first set would be the Q vectors because it uses a finite set of real numbers and inserts them into the extended vector. This is repeated to redefine a codebook for a certain dimensionality
- **there is provided a second set consisting of codevectors of dimension n+i that may not be obtained by insertion into the codevectors of the initial**

dictionary of the elements of said finite set with said collection of insertion rules

- [Gersho, Fig. 7], as described above shows the development of an extended vector of increased dimensionality by combining the Q vector and the S vector to produce the overall extended vector. In this case the S vector (estimated SSV) is not obtained by the predetermined rules. It is generated by a DFT (discrete Fourier transform) and further comprises the extended vector.
- **an intermediate dictionary, of dimension n+i comprising a union of said first set and of said second set is constructed, and steps a1) to a3) are repeated, at most N-n-1 times, with said intermediate dictionary in the guise of initial dictionary, up to said given dimension N**
- [Gersho, Fig. 7] discloses the extended vector X which is a union of the Q and S vectors and would be repeated to redefine a codebook for a certain dimensionality. The repetition is performed by using a placement vector Q which is an obvious variant of performing the insertion through iteration as described in the claim.

As per claim 18, claim 4 is incorporated and Gersho teaches:

- **an initial dictionary of initial resolution r.sub.n, higher than said given resolution r.sub.N, is obtained**
- [Gersho, Fig. 1] discloses that the given input vector is higher in dimensionality that the initial codebook. The resolution is dependent on the dimensionality of

the codebook and the number of codevectors in the codebook. The number of codevectors in the codebook does not change, so the resolution in this case is inversely related to the dimensionality. If the initial dictionary is lower in dimension than the given vector dimension, then the initial dictionary has a higher resolution than the given input vector.

- **on the basis of the initial dictionary, an intermediate dictionary of resolution r.sub.n-1 lower than the initial resolution r.sub.n, is constructed by partitioning of the initial dictionary into several subsets ordered according to a predetermined criterion**
- As shown in claim 5, the dimension is iteratively increased by insertion rules. The increase in dimension is a decrease in resolution. As shown in the rejection of claim 5, [Gersho, Fig. 7] shows how the Q and S vectors are used to create the higher in dimension vector X (which is also lower in resolution). The initial dictionary resolution is known and the intermediate dictionary, with lower resolution, is constructed by the partitioning of S and Q according to predetermined criterion in how the Q vector is comprised.
- **operation c'1) is repeated until the given resolution r.sub.N is attained**
- The entire operation is performed for all codevectors in the codebook such that a uniform resolution is obtained as shown in [Gersho, column 8, lines 29-36], “Equivalently, the encoder operation can be performed by constructing a new "codebook" by sub-sampling the universal codebook using Q to form a new set

of codevectors called subcodevectors, having the same dimensionality L.sub.Q as the input variable dimension vector."

As per claim 19, claim 18 is incorporated and Gersho teaches:

- **said predetermined criterion is chosen from among the cardinal of the subsets, an invoking of the subsets in a learning sequence, a contribution of the subsets to a total distortion or preferably to a decrease of this distortion**
- [Gersho, abstract] discloses "To complete the formulation, we define the distortion measure between an input SSV S with its associated selector vector Q and a spectral shape code vector Y.sub.j in the universal codebook. This measure is based on matching the input SSV samples to the corresponding subset of components of the spectral shape code vector Y.sub.j." The predetermined criterion is represented by the Q vector which is established by a distortion measure. It would be obvious to someone of ordinary skill in the art that the distortion measure would be minimized for the Q and spectral shape code vector's to be chosen, which teaches a decrease of distortion.

As per claim 20, claim 18 is incorporated and Gersho teaches:

- **said partition uses part at least of said insertion/deletion rules**
- As shown in claim 5, the dimension is iteratively increased by insertion rules. The increase in dimension is a decrease in resolution. As shown in the

rejection of claim 5, [Gersho, Fig. 7] shows how the Q and S vectors are used to create the higher in dimension vector X (which is also lower in resolution). The initial dictionary resolution is known and the intermediate dictionary, with lower resolution, is constructed by the partitioning of S and Q according to predetermined criterion in how the Q vector is comprised. Furthermore, it has been shown that the partitioning and combination of S and Q to develop the extended vector are based on the rules as shown in Fig. 7

As per claim 22, claim 4 is incorporated and Gersho teaches:

- **in step a), to attain the given dimension N, a first set consisting of codevectors formed by inserting/deleting, into/from codevectors of the first dictionary of dimension N' lower/higher than said given dimension N elements taken from a finite set of real numbers according to a finite collection of predetermined insertion/deletion rules is constructed**
- [Gersho, Fig. 7] discloses an example of how Gersho develops an extended vector by inserting zeros from the Q vector to change the dimensionality of the codeword vector. This is used in unison with the original vector to create the extended vector. The first set would be the Q vector because it uses a finite set of real numbers and inserts them into the extended vector. This is repeated to redefine a codebook for a certain dimensionality using the predetermined insertion rules using the Q vector.

- **in step b), subsequent to a possible step of definitive adaptation to the resolution r.sub.N, a second, definitive, dictionary comprising at least said first set is constructed for said given dimension N**
- [Gersho, column 8, lines 29-36] teaches "***Equivalently, the encoder operation can be performed by constructing a new "codebook***" by sub-sampling the universal codebook using Q to form a new set of codevectors called subcodevectors, having the same dimensionality L.sub.Q as the input variable dimension vector. Then, the encoder selects the subcodevector from this new codebook that best matches the input subvector." A new codebook is eventually established for a given dimension based on at least one resolution.

Gersho, fails to teach:

- **in step c), a first, intermediate, dictionary still of dimension N' but of higher/lower resolution r.sub.N is constructed on the basis of an initial dictionary of resolution r.sub.n and of dimension N' by embedding/simplification of dictionaries of increasing/decreasing resolutions, so as to substantially attain the resolution r.sub.N of said first dictionary**

Chan, in analogous art, teaches the above limitation,

- [Chan, page 3598, column 1] discloses "if the features have the same dimension and the same or similar statistics, then they can jointly share a codebook. ***For variable rate coding, the shared codebook must also be an embedded or multi-resolution codebook*** with which different degrees of

resolution can be attained to satisfy some bit allocation among the features." It would be obvious to embed/simplify dictionaries of different resolutions using the methods as disclosed in Gersho because Chan teaches that in variable rate encoding it would be inherent that the codebook would have to be embedded or multi-resolution.

- Chan and Gersho are analogous art because both pertain to variable dimension vector quantization. It would be obvious to someone of ordinary skill in the art to combine Chan with the Gersho device because [Chan, page 3597, column 1] discloses "We demonstrate that with CSVQ, tree structured codebooks can be constructed for very high rates without incurring an exponential growth in storage complexity and without impairing the rate-distortion performance. NLIVQ allows efficient coding of the power envelope, needed for transform coefficient normalization and adaptive distortion assignment. These techniques lead to a substantial reduction in the overall bit rate and codebook storage for the audio coder." Chan provides a method for reducing storage complexity in vector quantization analogous to the instant application.

As per claim 23, claim 4 is incorporated and Gersho discloses:

- **said second set consisting of codevectors that may not be obtained by application of the insertion/deletion to codevectors of lower/higher dimension than the given dimension according to said collection of insertion/deletion rules**

- [Gersho, Fig. 7], as described above shows the development of an extended vector of increased dimensionality by combining the Q vector and the S vector to produce the overall extended vector. In this case the S vector (estimated SSV) is not obtained by the predetermined rules. It is generated by a DFT (discrete Fourier transform) and further comprises the extended vector.
- **as well as at least one correspondence table making it possible to reconstitute any codevector of the dictionary of given dimension, using the indices of the insertion/deletion rules and indices identifying elements of said second set, thereby making it possible to avoid the complete storage of the dictionary for said given dimension, by simply storing the elements of said second set and links in the correspondence table for access to these elements and to the associated insertion/deletion rules**
- It would be obvious to someone of ordinary skill in the art that it would be implemented in such a fashion that pointers would be used to reference indices instead of copying the information to reduce memory requirements. Pointers are well-known in the art in languages such as C++ that tables can be defined and referenced by index instead of recalling the memory so that memory is conserved.

As per claim 24, claim 23 is incorporated and Gersho does not fully disclose:

- **a current dimension (j') of said second set**
- **a current index (m') of an element of the second set**

- **an insertion/deletion rule index (I.sub.r), this insertion/deletion rule at least contributing to reconstitute said codevector (x.sub.j) of the dictionary (D.sup.j.sub.Nj) of given dimension (j), by applying the insertion/deletion to the element of said current index (m') and of said current dimension (j').**
- It would be obvious to someone of ordinary skill in the art to characterize a codevector by index and index of its elements in a matrix. Furthermore, it would be obvious to someone of ordinary skill in the art to have a ruleset stored in one place that can be reached by a representative index of the wanted rule. Then, as described above, pointers could point from one index to another to gain the wanted result without actually storing the intermediary steps in order to effectively apply a rule to a codevector. Gersho discloses the use of a second set and rules to apply to that set where it would be obvious to someone of ordinary skill to use pointers, as in C++, to point to locations representing values and operations instead of actively defining them for every location for the purposes of lowering the storage requirements.

Claim 30 is rejected under the same principles as claim 4 for having parallel limitations. Although claim 30 further defines a computer program product, [Gersho, column 11, lines 58-60] discloses “It can be easily integrated with other structured VQ approaches to customize the encoding/decoding to the need of the application in terms of complexity, memory, performance targets.” It would be obvious that if Gersho

mentioned a memory that it can be assumed that Gersho's method can be embodied in a hardware embodiment or software embodiment with computer readable memory.

Claims 12 is considered to be unpatentable over Gersho et al. (US Patent #5890110 hereinafter Gersho) in view of Chan et al. (NPL document “Constrained-Storage Vector Quantization in High Fidelity Audio Transform Coding” hereinafter Chan) and further in view of Nishiguchi et al. (US Patent #5765127 hereinafter Nishiguchi).

As per claim 12, claim 10 is incorporated and Gersho fails to disclose:

- **said finite set is chosen by estimation of a monodimensional probability density of said source**

Nishiguchi, in analogous art, teaches the above limitation,

- [Nishiguchi, column 50-51, lines 65-67, 1] discloses "Originally, it is difficult to perform concrete design the codebook of the vector quantizer without knowing the distortion measure and the probability density function (PDF) of the input data." It would be obvious to someone of ordinary skill in that art that a finite set determined for the purpose of increasing/decreasing the codevectors in the codebook would be taught by Nishiguchi because a concrete design for a codebook would Involve determining the real numbered boundaries of the codebook for the dimension. A finite set for all dimensions lower than a maximum dimension could be calculated using a PDF of the input data.

Claims 13-14 are considered to be unpatentable over Gersho et al. (US Patent #5890110 hereinafter Gersho) in view of Chan et al. (NPL document "Constrained-Storage Vector Quantization in High Fidelity Audio Transform Coding" hereinafter Chan) and further in view of Bahl et al. (US Patent #5182773 hereinafter Bahl).

As per claim 13, claim 4 is incorporated and Gersho fails to fully teach:

- **said finite set and said collection of insertion/deletion rules are defined a posteriori after construction of dictionaries by embedding/simplification**

of dictionaries of successive resolutions, followed by a statistical analysis of these dictionaries thus constructed

Bahl, in analogous art, teaches the above limitations,

- [Bahl, column 4, lines 35-38] discloses “In a speech recognition system, the feature vectors extracted from the utterance of a speaker are labeled as different classes of sound in a process referred to as vector quantization…” Then further goes on to discloses in [Bahl, column 12, lines 13-15], "from a probabilistic point of view, the goal is to find the class of sound c with the highest a posteriori probability given an input speech vector x.” It would be obvious to someone of ordinary skill in the art that the finite set and collection of rules disclose in Gersho could be combined with the a posteriori analysis of the vectors as shown in Bahl.
- Bahl and Gersho are analogous art because both pertain to vector quantization. It would be obvious to someone of ordinary skill in the art to combine Bahl with the Gersho device because Bahl provides a posteriori analysis of vector quantization to " enhance the operation of a speech recognition system by eliminating, by possibly a factor of two to three, the error rate of the prior art technique [Bahl, column 3, lines 5-7].” Bahl provides the a posteriori analysis of vectors which would include defining features of the vectors. The feature interpretation can be incorporated in a finite set and collection of rules as in Gersho for removal of less important features to construct the dictionaries of successive resolutions.

As per claim 14, claim 10 is incorporated and Gersho teaches:

- **a first set and a first collection of insertion/deletion rules are chosen a priori by analysis of a learning sequence, so as to form one or more intermediate dictionaries**
- [Gersho, column 7, lines 49-55] discloses how the vector Q is defined, which teaches the insertion rules because it determines where the values are inserted into the extended vector X. Q is defined by the input pair of an estimated pitch value and a subvector and is done prior to constructing the dictionary. The updated dictionary is formed by this extension of vectors. It is intermediate because it must be iterated for all codevectors and is intermediate during the process.
- **as appropriate, at least one part of the set of codevectors forming said one or more intermediate dictionaries is also updated**
- This would be inherent as the codevectors are "extended" as shown above. The extension of the vectors updates them according to the rules.

Gersho fails to teach, but Bahl teaches in analogous art,

- **at least one part of said first set and/or of said first collection of insertion/deletion rules is updated by a posteriori analysis of said one or more intermediate dictionaries**
- [Bahl, column 4, lines 35-38] discloses “In a speech recognition system, the feature vectors extracted from the utterance of a speaker are labeled as

different classes of sound in a process referred to as vector quantization..." Then further goes on to discloses in [Bahl, column 12, lines 13-15], "from a probabilistic point of view, the goal is to find the class of sound c with the highest a posteriori probability given an input speech vector x." It would be obvious to someone of ordinary skill in the art that the finite set and collection of rules disclose in Gersho could be combined with the a posteriori analysis of the vectors as shown in Bahl. The rules define how the dictionary is built so it would be obvious that the rules would define an intermediate dictionary based on the iterate application of the rules.

- Bahl and Gersho are analogous art because both pertain to vector quantization. It would be obvious to someone of ordinary skill in the art to combine Bahl with the Gersho device because Bahl provides a posteriori analysis of vector quantization to " enhance the operation of a speech recognition system by eliminating, by possibly a factor of two to three, the error rate of the prior art technique [Bahl, column 3, lines 5-7]." Bahl provides the a posteriori analysis of vectors which would include defining features of the vectors. The feature interpretation can be incorporated in a finite set and collection of rules as in Gersho for removal of less important features to construct the dictionaries of successive resolutions.

Claims 28 is considered to be unpatentable over Gersho et al. (US Patent #5890110 hereinafter Gersho) in view of Kolesnik et al. (US Patent #5832443 hereinafter Kolesnik)

As per claim 28, claim 25 is incorporated and Gersho teaches:

- **CP4) an index of the rank of said nearest neighbour x.sup.j' in the dictionary D.sup.j.sub.i is determined**
- [Gersho, column 7, lines 8-11] discloses "The best codevector in this new codebook which matches the input vector, ***S is selected as the representative by the nearest neighbor block***, 803."

Gersho fails to teach:

- **CP1) on the basis of an input signal, an input vector y=(y.sub.0, . . . , y.sub.k, . . . , y.sub.j-1) defined by its absolute vector |y|=(|y.sub.0|, . . . ,|y.sub.k|, . . . ,|y.sub.j-1|) and by a sign vector .epsilon.=(.epsilon..sub.0, . . . ,.epsilon..sub.k, . . . ,.epsilon..sub.j-1) with .epsilon..sub.k=.+-.1 is formed,**
- **CP2) the components of the vector |y| are ranked by decreasing values, by permutation, to obtain a leader vector |{tilde over (y)}|**

Kolesnik, in analogous art, teaches the above limitations

- **CP1) on the basis of an input signal, an input vector y=(y.sub.0, . . . , y.sub.k, . . . , y.sub.j-1) defined by its absolute vector |y|=(|y.sub.0|, . . . ,|y.sub.k|, . . . ,|y.sub.j-1|) and by a sign vector .epsilon.=(.epsilon..sub.0, . . . ,.epsilon..sub.k, . . . ,.epsilon..sub.j-1) with .epsilon..sub.k=.+-.1 is formed,**
- [Kolesnik, column 4, lines 33-34] discloses "In step 124, a magnitude vector is created that comprises the absolute values of the selected transform coefficients." A magnitude vector teaches an absolute vector. Furthermore, [Kolesnik, column 4, lines 27-31] discloses "In step 122, a sign vector is created

identifying the signs of the selected transform coefficients. In one embodiment, the sign vector is a binary vector having ones in the relative locations of the positive coefficients and zeros in the relative locations of the negative coefficients." It would be obvious to someone of ordinary skill that binary values teach the +1, -1 relationship as an analogous step.

- Gersho and Kolesnik are analogous art because both pertain to vector quantization for compression. It would be obvious to someone of ordinary skill in the art to combine Kolesnik with the Gersho device because Kolesnik provides Gersho with a combinatorial coding technique to further reduce memory requirements. Kolesnik and Gersho are analogous inventions designed for the purposes of coding compression and decompression and elements that are analogous may be interchanged to produce analogous results.
- **CP2) the components of the vector |y| are ranked by decreasing values, by permutation, to obtain a leader vector |{tilde over (y)}|**
- [Kolesnik, column 4, lines 35-37] discloses "Using the magnitude vector, as well as a composition book and a quantization scale book, ***a rank vector*** and an indicator vector are also created in step 124." It would be obvious that the ranking by decreasing values is merely by decision.
- Gersho and Kolesnik are analogous art because both pertain to vector quantization for compression. It would be obvious to someone of ordinary skill in the art to combine Kolesnik with the Gersho device because Kolesnik provides Gersho with a combinatorial coding technique to further reduce memory

requirements. Kolesnik and Gersho are analogous inventions designed for the purposes of coding compression and decompression and elements that are analogous may be interchanged to produce analogous results.

- **CP3) a nearest neighbour x.sup.j' of the leader vector |{tilde over (y)}| is determined from among the leaders of the dictionary D.sup.j.sub.i of dimension j**
- Gersho has disclosed the use of a nearest neighbor technique as shown in [Gersho, column 7, lines 8-11], "The best codevector in this new codebook which matches the input vector, ***S is selected as the representative by the nearest neighbor block***, 803." This in conjunction with the ranked vector can be used to be compared to the absolute values in the codebook to reduce the memory by eliminating a negative sign. It would be an obvious step because it reduces memory and does not substantially alter the input vector set. It would be inherent that the leader vectors of the dictionary would have to be determined in advance or stored in the coder in advance with a corresponding relationship to the decoder.
- **CP5) and an effective value of coding/decoding is applied to the input vector, which is dependent on said index determined in step CP4), on said permutation determined in step CP2) and on said sign vector determined in step CP1).**
- Vector Quantization functionally words as the effective index of the matched codevector is applied to the input vector such that the input vector itself is not

transmitted or stored. This is dependent on what is determined to be the closest match between the codevector and the input vector as determine by the nearest neighbor algorithm disclosed in Gersho. It would be obvious that the permutation and the sign vector breakdown of the input vector would not affect how the effective value of coding or decoding is applied based on the index determined.

Allowable Subject Matter

As per claim 6, claim 4 is incorporated:

- **an initial dictionary of initial dimension n, higher than said given dimension N, is obtained**
- **a first set, of dimension n-i, is constructed by selection and extraction of possible codevectors of dimension n-i from the dictionary of dimension n, according to a finite collection of predetermined deletion rules**
- **there is provided a second set consisting of codevectors of dimension n-i, that may not be obtained by deletion, from the codevectors of the initial dictionary, of the elements of said finite set with said collection of deletion rules**
- **an intermediate dictionary, of dimension n-i comprising a union of said first set and of said second set is constructed, and steps a'1) to a'3) are repeated, at most n-N-1 times, with said intermediate dictionary in the guise of initial dictionary, up to said given dimension N**

Prior art shows codebooks of dimensionality lower than the given dimension which added elements to determine codevectors of varying dimension. However, the examiner was unable to find anything pertaining to an initial codebook with a higher dimension than the given dimension that further helped to determine possible codevectors by deleting elements in the initial codebook to obtain codevectors of each dimension to a dimension N.

As per claim 15, claim 4 is incorporated:

- **an initial dictionary of initial resolution r.sub.n, lower than said given resolution r.sub.N, is obtained**
- **on the basis of the initial dictionary, an intermediate dictionary of resolution r.sub.n+1 higher than the initial resolution r.sub.n is constructed**
- **operation c1) is repeated until the given resolution r.sub.N is attained**

The resolution is shown to depend on the dimensionality of the codebook and the number of codevectors in the codebook. The number of codevectors stays constant , thus the variable M stays constant while the dimensionality of the vector determines the resolution. Since the number of codevectors stay constant, the dimensionality is inversely related to the resolution and the initial dictionary in the above claim would have to have a higher dimension than the given dimension. This has been shown to be allowable in the subject matter of claim 6. Furthermore, claims 16, 17, and 21 are considered to contain allowable subject matter for their dependency on claim 15.

Conclusion

11. Refer to PTO-892, Notice of References Cited for a listing of analogous art.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to GREG A. BORSETTI whose telephone number is (571)270-3885. The examiner can normally be reached on Monday - Thursday (8am - 5pm Eastern Time).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Chameli Das can be reached on 571-272-3696. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Greg A. Borsetti/
Examiner, Art Unit 4141

/CHAMELI C. DAS/
Supervisory Patent Examiner, Art Unit 4141
Dated: 5/9/08